## IN THE CLAIMS:

Please amend claims 1, 5, 27, and 35 to better claim the invention.

1.     (Currently Amended) A method for operating a data storage system, comprising:

creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-encing changes in data stored in the data storage system after the writable vdisk was cre-ated, the writable vdisk having a plurality of holes where each hole is at locations refer-encing data which has not been changed since the writable vdisk was created;

maintaining a backing store, the backing store referencing data stored in the data storage system which has not been changed since the writable vdisk was created;

loading blocks of the writable vdisk from a disk into a memory, the loaded blocks including a writable vdisk indirect block having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a particular hole of the plu-rality of holes;

loading blocks of the backing store from a disk into the memory, the loaded blocks including a backing store indirect block having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, one or more backing store indirect block fields having a pointer to a data block;

searching each field of the writable vdisk indirect block for a hole; and

replacing each field having a hole in the writable vdisk indirect block with a new pointer to the data block referenced by the corresponding backing store indirect block field to update the writable vdisk to reference both the data which is unchanged since the writable vdisk was created and the data which has been changed since the writable vdisk was created-.

1    2. (Previously Presented) The method of claim 1, further comprising:

2          dirtying the data block pointed to by the backing store indirect block to enable

3   write allocation of the dirty data block without altering a data content of the data block.


1    3. (Previously Presented) The method of claim 1, further comprising:

2          choosing a new pointer for a newly allocated data block containing the  an unal-

3   tered data content;

4          setting bits in block allocation structures for the newly allocated data block; and

5          placing the new pointer to the newly allocated data block into the field of the wri-

6   table vdisk indirect block to replace the hole.


1    4. (Previously Presented) The method of claim 3 further comprising:

2          freeing the dirty data block; and

3          writing the newly allocated data block to disk.


1    5. (Currently Amended) The method of claim 4 further comprising ~~the~~:

2          releasing an association of the writable vdisk to the backing store to thereby

3          separate the writable vdisk data blocks from the backing store data blocks.


1    6. (Original) The method of claim 1 wherein the pointers contained in the writable vdisk

2   indirect block fields and the backing store indirect block fields comprise_logical volume

3   block numbers (VBNs).


1    7. (Original) The method of claim 1 wherein the invalid pointers contained in the wri-

2   table vdisk indirect block fields comprise a zero logical volume block number (VBN).

1    8. (Original)  The method of claim 1 wherein the plurality of fields in the writable vdisk

2    indirect block are a writable vdisk level 1 buffer and the plurality of fields in the backing

3    store indirect block are a backing store level 1 buffer.


1    9.  (Previously Presented)  An apparatus for operating a computer data base, comprising:

2         a writable virtual disk (vdisk) created at a selected time, the writable vdisk refer-

3    encing changes in data stored in the data storage system after the writable vdisk was cre-

4    ated, the writable vdisk having a plurality of holes where each hole is at locations refer-

5    encing data which has not been changed since the writable vdisk was created;

6         a backing store, the backing store referencing data stored in the data storage sys-

7    tem which has not been changed since the writable vdisk was created;

8         a backdoor message handler adapted to load blocks of the writable vdisk and

9    backing store from disk  into a memory of the storage system;

10        a writable vdisk indirect block in the memory having a plurality of fields, each

11   field storing a valid pointer to a data block or an invalid pointer representing a   particular

12   hole  of the plurality of holes;

13        a backing store indirect block in the memory having a plurality of fields, each

14   backing store indirect block field corresponding to a field of the writable vdisk indirect

15   block, each backing store indirect block field having a pointer to a data block;

16        a special loading function for searching each field of the writable vdisk indirect

17   block for one or more fields representing a hole; and

18        a write allocator for replacing each field representing a hole in the writable vdisk

19   indirect block with a new pointer to the data referenced by the corresponding backing

20   store indirect block field to update the writable vdisk to reference both the data which is

21   unchanged since the writable vdisk was created and the data which has been changed

22   since the writable vdisk was created.

1    10. (Previously Presented) The apparatus of claim 9 wherein the write allocator further

2    comprises:

3        a new pointer for a newly allocated data block containing an unaltered data content,

4    set bits in block allocation structures for the newly allocated data block, and place the

5    new pointer to the newly allocated data block into the field of the writable vdisk indirect

6    block to replace the hole.

1    11. (Original) The apparatus of claim 10 wherein the write allocator is further adapted

2        to:

3        free the dirty data block and write the newly allocated data block to disk.

1    12. (Original) The apparatus of claim 9 wherein the backdoor message handler loads the

2    blocks of the writable vdisk and the blocks of the backing store during periods of reduced

3    processing activity.

1    13. (Original) The apparatus of claim 9 wherein the pointers contained in the writable

2    vdisk indirect block fields and the backing store indirect block fields comprise logical

3    volume block numbers (VBNs).

1    14. (Original) The apparatus of claim 9 wherein the invalid pointers contained in the wri-

2    table vdisk indirect block fields comprise a zero logical volume block number (VBN).

1    15. (Original) The apparatus of claim 9 wherein the plurality of fields in the writable

2    vdisk indirect block comprises a writable vdisk level 1 buffer and the plurality of fields in

3    the backing store indirect block comprises a backing store level 1 buffer.

1    16 – 18 (Cancelled)

1    19. (Previously Presented) A data storage system apparatus, comprising:

5

2          means for creating a writable virtual disk (vdisk) at a selected time, the writable

3   vdisk referencing changes in data stored in the data storage system after the writable

4   vdisk was created, the writable vdisk having a plurality of holes where each hole is at lo-

5   cations referencing data which has not been changed since the writable vdisk was cre-

6   ated;

7          means for maintaining a backing store, the backing store referencing data stored

8   in the data storage system which has not been changed since the writable vdisk was cre-

9   ated;

10         means for loading blocks of the writable vdisk from a disk into a memory, the

11   loaded blocks including a writable vdisk indirect block having a plurality of fields, each

12   field storing a valid pointer to a data block or an invalid pointer representing a particular

13   hole of the plurality of holes;

14         means for loading blocks of the backing store from a disk into the memory, the

15   loaded blocks including a backing store indirect block having a plurality of fields, each

16   backing store indirect block field corresponding to a field of the writable vdisk indirect

17   block, one or more backing store indirect block fields having a pointer to a data block;

18         means for searching each field of the writable vdisk indirect block for a hole; and

19         means for replacing each field having a hole in the writable vdisk indirect block

20   with a new pointer to the data block referenced by the corresponding backing store indi-

21   rect block field to update the writable vdisk to reference both the data which is un-

22   changed since the writable vdisk was created and the data which has been changed since

23   the writable vdisk was created .


1   20. (Previously Presented) A computer readable medium, including program instruc-

2   tions executing on a computer, the program instructions including instructions for per-

3   forming the steps of:

4          creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-

5   encing changes in data stored in the data storage system after the writable vdisk was cre-

6 ated, the writable vdisk having a plurality of holes where each hole is at locations refer-

7 encing data which has not been changed since the writable vdisk was created;

8

9      maintaining a backing store, the backing store referencing data stored in the data

10 storage system which has not been changed since the writable vdisk was created;

11

12      loading blocks of the writable vdisk from a disk into a memory, the loaded blocks

13 including a writable vdisk indirect block having a plurality of fields, each field storing a

14 valid pointer to a data block or an invalid pointer representing a particular hole of the plu-

15 rality of holes;

16      loading blocks of the backing store from a disk into the memory, the loaded

17 blocks including a backing store indirect block having a plurality of fields, each backing

18 store indirect block field corresponding to a field of the writable vdisk indirect block, one

19 or more backing store indirect block fields having a pointer to a data block;

20      searching each field of the writable vdisk indirect block for a hole; and

21      replacing each field having a hole in the writable vdisk indirect block with a new

22 pointer to the data block referenced by the corresponding backing store indirect block

23 field to update the writable vdisk to reference both the data which is unchanged since the

24 writable vdisk was created and the data which has been changed since the writable vdisk

25 was created .


21 – 22  (Cancelled)


1 23. (Previously Presented) A method for operating a data storage system, comprising:

2      creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-

3 encing changes in data stored in the data storage system after the writable vdisk was cre-

4 ated, the writable vdisk having a plurality of holes where each hole is at a location refer-

5 encing data which has not been changed since the writable vdisk was created;

6    maintaining a backing store, the backing store referencing the data stored in the

7    data storage system which has not been changed since the writable vdisk was created;

8    searching each field of the writable vdisk for a hole; and

9    referencing each hole in the writable vdisk to point to the data block referenced by

10   the corresponding backing store indirect block to update the writable vdisk to reference

11   both the data which is unchanged since the writable vdisk was created and the data which

12   has been changed since the writable vdisk was created.


1    24. (Previously Presented)  The method of claim 23, further comprising:

2    dirtying the data block pointed to by the backing store indirect block to enable

3    write allocation of the dirty data block without altering a data content of the data block.


1    25. (Previously Presented)  The method of claim 23 further comprising:

2    choosing a new pointer for a newly allocated data block containing the  an unal-

3    tered data content;

4    setting bits in block allocation structures for the newly allocated data block; and

5    placing the new pointer to the newly allocated data block into the field of the wri-

6    table vdisk indirect block to replace the hole.


1    26. (Previously Presented)  The method of claim 25, further comprising:

2    freeing the dirty data block; and

3    writing the newly allocated data block to disk.


1    27. (Currently Amended)  The method of claim 26 further comprising the:

2    releasing an association of the writable vdisk to the backing store to thereby

3    separate the writable vdisk data blocks from the backing store data blocks.


1    28. (Previously Presented)  The method of claim 23, further comprising:

2  including logical volume block numbers (VBNs) in the pointers contained in the

3  writable vdisk indirect block fields and the backing store indirect block fields.

1  29. (Previously Presented)  The method of claim 23, further comprising:

2    using a zero logical volume block number (VBN) as the invalid pointers con-

3  tained in the writable vdisk indirect block fields.

1  30. (Previously Presented)  The method of claim 23, further comprising:

2    using a writable vdisk level 1 buffer for the plurality of fields in the writable vdisk

3  indirect block and using a backing store level 1 buffer for the plurality of fields in the

4  backing store indirect block.

1  31.  (Previously Presented)  A data storage system, comprising:

2    a writable virtual disk (vdisk) created at a selected time, the writable vdisk refer-

3  encing changes in data stored in the data storage system after the writable vdisk was cre-

4  ated, the writable vdisk having a plurality of holes where each hole is at a location refer-

5  encing data which has not been changed since the writable vdisk was created;

6    a backing store, the backing store referencing the data stored in the data storage

7  system which has not been changed since the writable vdisk was created;

8    a processor to search each field of the writable vdisk for a hole; and

9    the processor to reference each hole in the writable vdisk to point to the data

10  block referenced by the corresponding backing store indirect block to update the writable

11  vdisk to reference both the data which is unchanged since the writable vdisk was created

12  and the data which has been changed since the writable vdisk was created.

1  32. (Previously Presented)  The system of claim 31, further comprising:

2    the data block pointed to by the backing store indirect block are dirtied to enable

3  write allocation of the dirty data block without altering a data content of the data block.

1   33. (Previously Presented) The system of claim 31 further comprising:

2       a new pointer chosen for a newly allocated data block containing an unaltered

3   data content;

4       bits are set in a block allocation structures for the newly allocated data block; and

5       a new pointer to the newly allocated data block placed into a field of the writable

6   vdisk indirect block to replace the hole.


1   34. (Previously Presented) The system of claim 33, further comprising:

2       the dirty data block is freed; and

3       the newly allocated data block is written to disk.


1   35. (Currently Amended) The system of claim 34 further comprising the:

2       an association of the writable vdisk to the backing store is released to thereby

3   separate the writable vdisk data blocks from the backing store data blocks.


1   36. (Previously Presented) The system of claim 31, further comprising:

2       logical volume block numbers (VBNs) included in the pointers contained in the

3   writable vdisk indirect block fields and the backing store indirect block fields.


1   37. (Previously Presented) The system of claim 31, further comprising:

2       a zero logical volume block number (VBN) used as the invalid pointers contained

3   in the writable vdisk indirect block fields.


1   38. (Previously Presented) The system of claim 31, further comprising:

2       a writable vdisk level 1 buffer used for the plurality of fields in the writable vdisk

3   indirect block and a backing store level 1 buffer used for the plurality of fields in the

4   backing store indirect block.

1    39. (Previously Presented) A computer readable media, comprising:

2        said computer readable media containing instructions for execution on a processor

3    for a method of method for operating a data storage system, the method having,

4        creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-

5    encing changes in data stored in the data storage system after the writable vdisk was cre-

6    ated, the writable vdisk having a plurality of holes where each hole is at a location refer-

7    encing data which has not been changed since the writable vdisk was created;

8        maintaining a backing store, the backing store referencing the data stored in the

9    data storage system which has not been changed since the writable vdisk was created;

10        searching each field of the writable vdisk for a hole; and

11        referencing each hole in the writable vdisk to point to the data block referenced by

12    the corresponding backing store indirect block to update the writable vdisk to reference

13    both the data which is unchanged since the writable vdisk was created and the data which

14    has been changed since the writable vdisk was created.

Please add claims 40 *et seq.*,

1    40. (New)  A method for operating a data storage system, comprising:

2          creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-

3    encing changes in data stored in the data storage system after the writable vdisk was cre-

4    ated, the writable vdisk having a plurality of holes where each hole is at a location refer-

5    encing data which has not been changed since the writable vdisk was created;

6          maintaining a backing store, the backing store referencing the data stored in the

7    data storage system which has not been changed since the writable vdisk was created;

8          searching, by a background task process, each field of the writable vdisk for a

9    hole; and

10         referencing each hole in the writable vdisk to point to the data block referenced by

11   the corresponding backing store indirect block to update the writable vdisk to reference

12   both the data which is unchanged since the writable vdisk was created and the data which

13   has been changed since the writable vdisk was created.


1    41. (New)  A data storage system, comprising:

2          a writable virtual disk (vdisk) created at a selected time, the writable vdisk refer-

3    encing changes in data stored in the data storage system after the writable vdisk was cre-

4    ated, the writable vdisk having a plurality of holes where each hole is at a location refer-

5    encing data which has not been changed since the writable vdisk was created;

6          a backing store, the backing store referencing the data stored in the data storage

7    system which has not been changed since the writable vdisk was created;

8          a background task_processor to search each field of the writable vdisk for a hole;

9    and

10         the processor to reference each hole in the writable vdisk to point to the data

11   block referenced by the corresponding backing store indirect block to update the writable

12   vdisk to reference both the data which is unchanged since the writable vdisk was created

13   and the data which has been changed since the writable vdisk was created.

1   42. (New) A computer readable media, comprising:

2       said computer readable media containing instructions for execution on a processor

3   for a method of method for operating a data storage system, the method having,

4       creating a writable virtual disk (vdisk) at a selected time, the writable vdisk refer-

5   encing changes in data stored in the data storage system after the writable vdisk was cre-

6   ated, the writable vdisk having a plurality of holes where each hole is at a location refer-

7   encing data which has not been changed since the writable vdisk was created;

8       maintaining a backing store, the backing store referencing the data stored in the

9   data storage system which has not been changed since the writable vdisk was created;

10       searching, by a background task process, each field of the writable vdisk for a

11   hole; and

12       referencing each hole in the writable vdisk to point to the data block referenced by

13   the corresponding backing store indirect block to update the writable vdisk to reference

14   both the data which is unchanged since the writable vdisk was created and the data which

15   has been changed since the writable vdisk was created.